Smoothing Dynamic Systems with State-Dependent Covariance Matrices

Aleksandr Aravkin
Departments of Computer Science and EOS
University of British Columbia, Vancouver, BC
saravkin@eos.ubc.ca

James Burke
Department of Mathematics
University of Washington, Seattle, WA
jvburke@uw.edu

Abstract

Kalman filtering and smoothing algorithms are used in many areas, including tracking and navigation, medical applications, and financial trend filtering. One of the basic assumptions required to apply the Kalman smoothing framework is that error covariance matrices are known and given. In this paper, we study a general class of inference problems where covariance matrices can depend functionally on unknown parameters. In the Kalman framework, this allows modeling situations where covariance matrices may depend functionally on the state sequence being estimated. We present an extended formulation and novel algorithm for inference in this context, specify these to the Kalman smoothing case, and develop an efficient implementation that preserves the computational efficiency of the Kalman smoother. The method is illustrated with a synthetic numerical example.

1 Introduction

The Kalman filter [14] and smoother [17] are efficient algorithms to estimate the state of a dynamic system given noisy measurements. Over the last 10 years, the optimization perspective on the smoothing problem has produced many extensions to dynamic system estimation, including methods for smoothing systems with nonlinear process and measurement models [7], systems with nonlinear inequality constraints [9], robust Kalman smoothing [5, 4, 3], and smoothing of sparse systems [1].

In all of the above extensions, the variances of process and measurement errors are assumed to be fixed and known. In practice, these quantities are often not known, and may in fact depend on the state. For example, radar position errors are known to depend on the aspect angle as well as the position of the target [18]. In some applications [15], it may be of interest to do Kalman filtering in polar coordinates or other coordinates that induce a state dependence in measurement errors. Modeling of process error covariance may also be state dependent — for example, Bar-Shalom [6] suggests that the right choice of process noise level for flight tracking models depends on the turn rate range expected. Therefore if we are estimating turn rate as part of the state, the process noise level can be modeled as a function of (a portion of) the state.

These ideas motivate extensions of the standard Kalman smoothing formulation to situations where process and measurement variances have known functional dependence on the state. Several such extensions have already been considered. In [18], the Unscented Transform is used to fit models with state-dependent matrices acting on observation noise. Linear systems with additive observation noise where measurement error variance is a known function of the state are studied in [19]. Linear systems with control inputs transformed by state-dependent matrices are considered in [12]. Finally, adaptive system identification, as presented in [11], also falls into this class.

In this paper, we formulate the state-dependent covariance problem as a statistical estimation problem, and develop algorithms for obtaining the *maximum a posteriori* (MAP) estimate. In the theoretical development, we allow the process and measurement functions to be nonlinear, and we allow the functional dependence of covariance on the state to be nonlinear as well.

The paper proceeds as follows. In Section 2, we formulate the Kalman model with state-dependent covariances and develop the MAP objective to be optimized. While this objective is motivated by the Kalman smoother, it can be used for general nonlinear regression where variance depends in a known functional way on the parameters. In Section 3 we build a new algorithm for solving the resulting optimization problems that exploits their convex composite structure. In Section 4 we provide a numerical experiment using simulated data that demonstrates the performance of the new smoother and the potential modeling capabilities of the approach. Further algorithmic details for the Kalman smoothing problem are explained in the Appendix.

2 Kalman Smoothing with State-Dependent Uncertainty

The dynamic structure of the Kalman smoothing problem is specified as follows:

$$\begin{array}{rcll} \mathbf{x_1} &=& g_1(x_0) + \mathbf{w_1}, & \\ \mathbf{x_k} &=& g_k(\mathbf{x_{k-1}}) + \mathbf{w_k} & k = 2, \ldots, N, \\ \mathbf{z_k} &=& h_k(\mathbf{x_k}) + \mathbf{v_k} & k = 1, \ldots, N \ , \end{array} \tag{1}$$

where g_k, h_k are known (nonlinear) process and measurement functions, and $\mathbf{w_k} \in \mathbb{R}^n$, $\mathbf{v_k} \in \mathbb{R}^{m(k)}$ are mutually independent Gaussian random variables with positive definite covariance matrices Q_k and R_k, $\mathbf{x_k} \in \mathbb{R}^n$ are the unknown states, and $\mathbf{z_k} \in \mathbb{R}^{m(k)}$ are the observed measurements. We remove the assumption that Q_k and R_k are fixed and known, and instead model these covariance matrices as $\mathcal{C}^2$ functions of the state.

In order to develop a simpler notation for estimating the entire state sequence, we define functions $g : \mathbb{R}^{nN} \to \mathbb{R}^{nN}$ and $h : \mathbb{R}^{nN} \to \mathbb{R}^M$, with $M = \sum_k m_k$, from components g_k and h_k as follows:

$$g(x) = \begin{bmatrix} x_1 \\ x_2 - g_2(x_1) \\ \vdots \\ x_N - g_N(x_{N-1}) \end{bmatrix} , \quad h(x) = \begin{bmatrix} h_1(x_1) \\ h_2(x_2) \\ \vdots \\ h_N(x_N) \end{bmatrix} . \tag{2}$$

Given a sequence of column vectors $\{u_k\}$ and matrices $\{T_k\}$ we use the following notation:

$$\mathrm{vec}(\{u_k\}) = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} , \ \mathrm{diag}(\{T_k\}) = \begin{bmatrix} T_1 & 0 & \cdots & 0 \\ 0 & T_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & T_N \end{bmatrix} , \quad \begin{array}{r} R = \mathrm{diag}(\{R_k\}) \\ Q = \mathrm{diag}(\{Q_k\}) \\ x = \mathrm{vec}(\{x_k\}) \\ w = \mathrm{vec}(\{g_0, 0, \ldots, 0\}) \\ z = \mathrm{vec}(\{z_1, \ldots, z_N\}) \end{array} \tag{3}$$

where $g_0 := g_1(x_0)$.

With this notation, and under Gaussian assumptions, the MAP object for the Kalman smoother is given by

$$-\log\det\left(Q^{-1/2}(x)\right) - \log\det\left(R^{-1/2}(x)\right) + \frac{1}{2}\|Q^{-1/2}(x)(g(x)-w)\|_2^2 + \frac{1}{2}\|R^{-1/2}(x)(h(x)-z)\|_2^2 \ . \tag{4}$$

3 Convex Composite Formulation

We apply the Gauss-Newton methodology for minimizing convex composite functions [10] to the objective (4). The first step in this process is to write this objective in convex composite form, that is, in the form $f = \rho \circ F$, where ρ is convex and F is smooth. The choice of the functions ρ and F depend on how we wish to model the representation of the problem. The most straightforward way to rewrite (4) is the more general form

$$J(x) := \frac{1}{2} c(x)^T W(x)^{-1} c(x) + \frac{1}{2} \log\det(W(x)),$$

where $c : \mathbb{R}^{nN} \to \mathbb{R}^{M+nN}$ and $W : \mathbb{R}^{nN} \to \mathcal{S}_{++}^{M+nN}$ are smooth, and $\mathcal{S}_{++}^{M+nN}$ is the cone of real symmetric $(M+nN) \times (M+nN)$ positive definite matrices. Then $J = \hat{\rho} \circ \hat{F}$ with

$$\hat{\rho}(c, W) := \frac{1}{2} c^T W^{-1} c + \frac{1}{2} \log\det(W) \quad \text{and} \quad \hat{F}(x) = (c(x), W(x)).$$

Although the function $\hat{F}$ in this formulation can be assumed smooth, the function $\hat{\rho}$ is not convex. Indeed, $\hat{\rho}$ is the difference of two convex functions. When viewed as a function of (c, W^{-1}), it is still not jointly convex in these arguments. Hence, in order to obtain a convex composite formulation, a different approach must be taken.

Here, we propose an approach that applies in many practical settings and yields an efficient solution procedure. However, a price is paid in a more complex model for the covariance matrices. Specifically, we assume that the Cholesky factors for $Q_k^{-1}(x_k)$ and $R_k^{-1}(x_k)$ are given to us as explicit functions of the state. We denote these factors by $Q_k^{-1/2}(x_k)$ and $R_k^{-1/2}(x_k)$, respectively. In some settings, the matrices $Q_k(x_k)$ and $R_k(x_k)$ are modeled as diagonal matrices, in which case the inverse Cholesky factors are easily computed diagonal matrices. We provide an example of this type in the final section. Under this modeling assumption, the objective (4) can be abstracted to the more general form

$$K(x) = \frac{1}{2}c(x)^{\mathrm{T}}V(x)^{T}V(x)c(x) - \log \circ \det[V(x)] \ , \tag{5}$$

where $c : \mathbb{R}^{nN} \to \mathbb{R}^{M+nN}$ and $V : \mathbb{R}^{nN} \to \mathcal{L}_{++}^{M+nN}$ are smooth, and $\mathcal{L}_{++}^{M+nN}$ is the cone of $(M+nN) \times (M+nN)$ real lower triangular matrices with strictly positive entries on the diagonal. Then K can be written in convex composite form $K(x) = \rho \circ F$ with

$$\rho(u, v) \quad = \quad \frac{1}{2}u^{\mathrm{T}}u - \sum_i \log[v_i] \ ; \quad F(x) = \begin{bmatrix} F_1(x) \\ F_2(x) \end{bmatrix} = \begin{bmatrix} V(x)c(x) \\ \mathrm{vec}[\{V_{ii}(x)\}] \end{bmatrix} \ . \tag{6}$$

The direction finding subproblem in a Gauss-Newton method takes the form

$$\min_d \rho(F(x) + F'(x)d) \ .$$

Linearizing the functions $F_i(x)$ in (6) yields approximations $\tilde{F}_i(x; d) := F_i(x) + F_i(x)d$, which in turn gives the approximation

$$\tilde{K}(x; d) = \rho[\tilde{F}_1(x; d), \tilde{F}_2(x; d)] \ . \tag{7}$$

This is the objective for the direction finding subproblem. Here

$$\begin{array}{rcl} \tilde{F}_1(x; d) & = & V(x)c(x) + \left(V(x)\partial_x c(x) + (c(x)^{\mathrm{T}} \otimes I_N)\partial_x V(x)\right) d \\ \tilde{F}_2(x; d) & = & \mathrm{vec}\left(\{V_{ii}(x) + \partial_x V_{ii}(x)d\}\right) . \end{array} \tag{8}$$

Note that we must be sure that $\tilde{F}_2(x; d)$ is component-wise greater than zero. For details of these derivations, see [2]. The Gauss-Newton subproblem is now given by

$$\min_{d \in \mathbb{R}^{nN}} \frac{1}{2}\tilde{F}_1(x; d)^{\mathrm{T}}\tilde{F}_1(x; d) - \sum_i \log[\tilde{F}_2(x; d)] \ . \tag{9}$$

If we ignore dependence on x, this problem can be rewritten as

$$\min_{d \in \mathbb{R}^{nN}} \frac{1}{2}d^{\mathrm{T}}Cd + a^{\mathrm{T}}d - \sum_i \log[s_i] \quad \text{s.t.} \quad s = \mathrm{vec}\{V_{ii}\} + \partial_x \mathrm{vec}\{V_{ii}\}d, \tag{10}$$

where

$$\begin{array}{rcl} C & = & [V\partial_x c + (c^{\mathrm{T}} \otimes I_N)\partial_x V]^{\mathrm{T}}[V\partial_x c + (c^{\mathrm{T}} \otimes I_N)\partial_x V], \\ a & = & c^{\mathrm{T}}V^{\mathrm{T}}V\partial_x c + c^{\mathrm{T}}V^{\mathrm{T}}(c^{\mathrm{T}} \otimes I_N)\partial_x V. \end{array} \tag{11}$$

Note that a is the gradient of the quadratic portion of the extended objective with respect to the state sequence x. The quantity $(c^{\mathrm{T}} \otimes I_N)\partial_x V$ that appears in (11) can be rewritten as

$$(c^{\mathrm{T}} \otimes I_N)\partial_x V \quad = \quad \textstyle\sum_{i=1}^{M+nN} c_i \partial_x V_i. \tag{12}$$

The Lagrangian associated with this problem is

$$L(d, s, \lambda) = \frac{1}{2}d^{\mathrm{T}}Cd + a^{\mathrm{T}}d - \sum_i \log[s_i] + \lambda^{\mathrm{T}}\left(s - \mathrm{vec}\{V_{ii}\} - \partial_x \mathrm{vec}\{V_{ii}\}d\right) . \tag{13}$$

The corresponding optimality conditions are

$$\begin{array}{rcl} \nabla_d L & = & Cd + a - \partial_x \mathrm{vec}\{V_{ii}\}^{\mathrm{T}}\lambda = 0 \\ \nabla_s L & = & -D(s)^{-1}\mathbf{1} + \lambda = 0 \\ \nabla_\lambda L & = & s - \mathrm{vec}\{V_{ii}\} - \partial_x \mathrm{vec}\{V_{ii}\}d = 0 \ . \end{array} \tag{14}$$

To solve the direction finding subproblem, we apply a damped Newton method directly to the optimality conditions. The details are given in section 5.1 of the Appendix. Explicit computations for the Kalman smoothing case are given in section 5.2 of the Appendix.

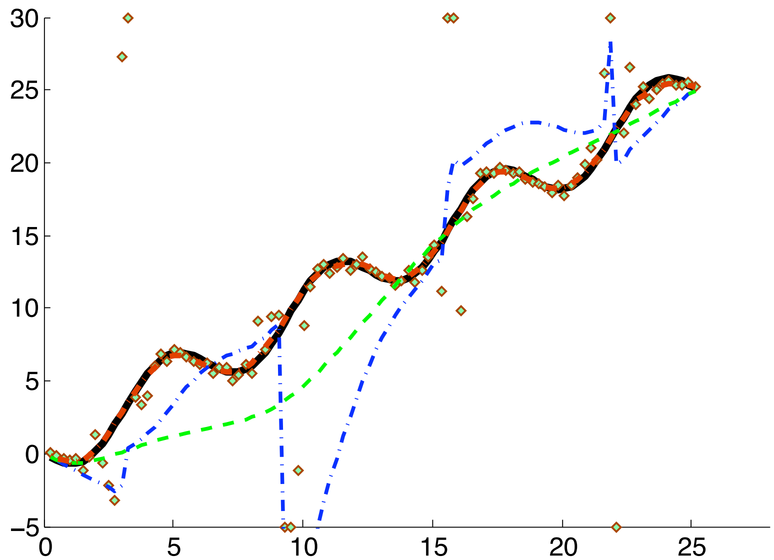

Figure 1: True state x_1 (black curve), Extended Smoother estimate (thick red dash-dot), Kalman filter estimate (blue dash-dot) and Kalman Smoother estimate (green dashed curve). Measurements are displayed as diamonds, and those outside the axis range are displayed on the figure boundary.

4 Numerical Results

In this section, we present some numerical experiments to show the advantages and modeling possibilities of the new Kalman smoother. The simulation model we consider is similar to the one presented in [9]. The 'ground truth' time series for this simulated example is given by

$$x(t) = \begin{bmatrix} 1 - 2\cos(t) \\ t - 2\sin(t) \end{bmatrix} .$$

The time between measurements is a constant denoted by Δt. The models for the mean of x_k given x_{k-1} and for process covariance Q_k [13, 16] are

$$g_k(x_{k-1}) = \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} x_{k-1} , \quad Q_k = \begin{bmatrix} \Delta t & \Delta t^2/2 \\ \Delta t^2/2 & \Delta t^3/3 \end{bmatrix} .$$

The measurement model for the mean of z_k given x_k is $h_k(x_k) = x_{2,k}$, where $x_{2,k}$ denotes the second component of x_k.

The main innovation of the example is in the measurement variance model. The smoother takes inverse Cholesky factors as input, and these are assumed to be $3 - x_{1,k}$. Then the variance model is given by $R_k(x_k) = (3 - x_{1,k})^{-2}$. The measurements were generated using the measurement model, from two full periods of the time series $x(t)$, with $N = 100$ discrete time points equally spaced over the interval $[0, 4\pi]$, and with noise sampled from $N(0, R_k(x_k))$. Since the true state x_1 varies in the interval $[-1, 3]$, the variance for the observations goes to infinity when t is a multiple of π.

This simulation illustrates a situation where the measurements are very reliable for some state values, but completely unpredictable for others. This phenomenon may occur for example if sensors report garbage values when the attitude of a vehicle is in a particular configuration. The measurement model presented here can be easily adapted by the user to take their beliefs about the system into account. The main point is that as long as the inverse Cholesky factors for the variance can be coded as a smooth function of the state, smoothed estimates for state values can be obtained taking into account this bad behavior of the measurements.

The result of the simulation is shown in Figure 1. The extended Kalman smoother (thick red dash-dot) is able to recover the ground truth (shown in black) with no appreciable difference. The Kalman filter (thin blue dash-dot) is strongly affected by the outlying measurements, as expected.

The Kalman smoother (green dashed) is able to smooth the measurements, but cannot pick up the oscillations of the ground truth, which are small in magnitude compared to the size of the errors.

This last point is the most important — it is not just the magnitude of the outliers that makes the Kalman smoother fail, although it can be seen to be rather far off the ground truth. The biggest challenge of the situation presented is knowing which measurements to trust, since this information depends on the state being estimated.

References

[1] D. Angelosante, S.I. Roumeliotis, and G.B. Giannakis. Lasso-kalman smoother for tracking sparse signals. In *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*, pages 181–185, nov. 2009.

[2] A.Y. Aravkin. *Robust Methods with Applications to Kalman Smoothing and Bundle Adjustment*. PhD thesis, University of Washington, Seattle, WA, June 2010.

[3] A.Y. Aravkin, B.M. Bell, J.V. Burke, and G. Pillonetto. An ℓ_1-laplace robust kalman smoother. *Automatic Control, IEEE Transactions on*, 56(12):2898–2911, dec. 2011.

[4] A.Y. Aravkin, J.V. Burke, and G. Pillonetto. Robust and Trend-following Kalman Smoothers using Students t . In *IFAC, 16th Symposium of System Identification*, oct. 2011.

[5] A.Y. Aravkin, J.V. Burke, and G. Pillonetto. A statistical and computational theory for robust and sparse Kalman smoothing. In *IFAC, 16th Symposium of System Identification*, oct. 2011.

[6] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, 2001.

[7] B.M. Bell. The iterated Kalman smoother as a Gauss-Newton method. *SIAM J. Optimization*, 4(3):626–636, August 1994.

[8] B.M. Bell. The marginal likelihood for parameters in a discrete Gauss-Markov process. *IEEE Transactions on Signal Processing*, 48(3):626–636, August 2000.

[9] B.M. Bell, J.V. Burke, and G. Pillonetto. An inequality constrained nonlinear kalman-bucy smoother by interior point likelihood maximization. *Automatica*, 45(1):25–33, January 2009.

[10] J.V. Burke. Descent methods for composite nondifferentiable optimization problems. *Mathematical Programming*, 33:260–279, 1985.

[11] C.K. Chui and G. Chen. *Kalman Filtering: with Real-Time Applications*. Springer series in information sciences. Springer, 2008.

[12] A. Dutka, H. Javaherian, and M.J. Grimble. State-dependent Kalman filters for robust engine control. In *Proceedings of the 2006 American Control Conference*, pages 1185–1190, 2006.

[13] Andrew Jazwinski. *Stochastic Processes and Filtering Theory*. Dover Publications, Inc, 1970.

[14] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the AMSE - Journal of Basic Engineering*, 82(D):35–45, 1960.

[15] G.A. McIntyre and K.J. Hintz. A comparison of several maneuvering target tracking models. In *SPIE Conference on Signal Processing, Sensor Fusion, and Target Recognition VII*, volume 3374, pages 48–63, April 1998.

[16] Bernt Oksendal. *Stochastic Differential Equations*. Springer, sixth edition, 2005.

[17] H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA J.*, 3(8):1145–1150, 1965.

[18] M. Stakkeland, O. Overrein, and E.F. Brekke. Tracking of targets with state dependent measurement errors using recursive BLUE filters. In *12th International Conference on Information Fusion*, pages 2052–2061, 2009.

[19] B. Zehnwirth. A generalization of the Kalman filter for models with state-dependent observation variance. *Journal of the Amer. Stat. Association*, 83(401):164–167, March 1988.

5 Appendix

5.1 Newton Iterations

$$
\begin{aligned}
E(s,\lambda,d) &= \begin{bmatrix} s - \mathrm{vec}\{V_{ii}\} - \partial_x \mathrm{vec}\{V_{ii}\} d \\ D(s)D(\lambda)\mathbf{1} - \mathbf{1} \\ Cd + a - \partial_x \mathrm{vec}\{V_{ii}\}^{\mathrm{T}}\lambda \end{bmatrix} &(15)\\
\nabla E(s,\lambda,d) &= \begin{bmatrix} I & 0 & -\partial_x \mathrm{vec}\{V_{ii}\} \\ D(\lambda) & D(s) & 0 \\ 0 & -\partial_x \mathrm{vec}\{V_{ii}\}^{\mathrm{T}} & C \end{bmatrix} &(16)
\end{aligned}
$$

To get the search direction for (10), we solve the equation

$$
\nabla E(s,\lambda,d) \begin{bmatrix} \Delta s \\ \Delta \lambda \\ \Delta d \end{bmatrix} = -E(s,\lambda,d). \tag{17}
$$

Using row operations $R_2 = R_2 - D(\lambda)R_1$ and $R_3 = R_3 + \partial_x \mathrm{vec}\{V_{ii}\}^{\mathrm{T}} D(s)^{-1} R_2$, we obtain the modified system

$$
\begin{bmatrix} I & 0 & -\partial_x \mathrm{vec}\{V_{ii}\} \\ 0 & D(s) & D(\lambda)\partial_x \mathrm{vec}\{V_{ii}\} \\ 0 & 0 & C + \partial_x \mathrm{vec}\{V_{ii}\}^{\mathrm{T}} D(s)^{-1} D(\lambda) \partial_x \mathrm{vec}\{V_{ii}\} \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta \lambda \\ \Delta d \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}, \tag{18}
$$

where

$$
\begin{aligned}
&\alpha = -s + \mathrm{vec}\{V_{ii}\} + \partial_x \mathrm{vec}\{V_{ii}\} d \\
&\beta = \mathbf{1} - D(\lambda)\left(\mathrm{vec}\{V_{ii}\} + \partial_x \mathrm{vec}\{V_{ii}\} d\right) \\
&\gamma = -Cd - a + \partial_x \mathrm{vec}\{V_{ii}\}^{\mathrm{T}} \left(\lambda + D(s)^{-1}\left(\mathbf{1} - D(\lambda)(\mathrm{vec}\{V_{ii}\} + \partial_x \mathrm{vec}\{V_{ii}\} d)\right)\right) .
\end{aligned}
$$

From here, we can recover the direction:

$$
\begin{aligned}
\Delta d &= \left(C + \partial_x \mathrm{vec}\{V_{ii}\}^{\mathrm{T}} D(s)^{-1} D(\lambda) \partial_x \mathrm{vec}\{V_{ii}\}\right)^{-1} \gamma \\
\Delta \lambda &= D(s)^{-1}\Big(\mathbf{1} - D(\lambda)(\mathrm{vec}\{V_{ii}\} + \partial_x \mathrm{vec}\{V_{ii}\}(d + \Delta d))\Big) \\
\Delta s &= -s + \mathrm{vec}\{V_{ii}\} + \partial_x \mathrm{vec}\{V_{ii}\}(d + \Delta d)
\end{aligned} \tag{19}
$$

Note that any damping scheme will require that we have $s > 0$ for the objective to be finite, and hence we will also have $\lambda > 0$.

5.2 Structure of the Extended Kalman Smoothing Objective

We now specify the method in the previous section to the Kalman smoothing problem, to demonstrate that the computational efficiency of the Kalman smoother can be preserved.

Define the matrix function $V(x)$ by

$$
V(x) = \mathrm{diag}\left(\{Q^{-1/2}(x), R^{-1/2}(x)\}\right) , \tag{20}
$$

with Q and R defined in (3). We also define the vector function $c(x)$ by

$$
c(x) = \begin{bmatrix} g(x) - w \\ h(x) - z \end{bmatrix}. \tag{21}
$$

with g and h as in (3).

With these definitions, objective (5) is exactly (4), and can be written explicitly as follows:

$$
\begin{aligned}
K(x) &= \tfrac{1}{2}\Big(c^{\mathrm{T}}(x)V(x)^{\mathrm{T}}V(x)c(x)\Big) - \log\circ\det[V(x)] \\
&= \tfrac{1}{2}\textstyle\sum_{k=1}^{N}[z_k - h_k(x_k)]^{\mathrm{T}} R_k^{-T/2}(x_k) R_k^{-1/2}(x_k)[z_k - h_k(x_k)] \\
&\quad + \tfrac{1}{2}\textstyle\sum_{k=0}^{N}[x_k - g_k(x_{k-1})]^{\mathrm{T}} Q_k^{-T/2}(x_k) Q_k^{-1/2}(x_k)[x_k - g_k(x_{k-1})] \\
&\quad - \log\det(R_k^{-1/2}(x_k)) - \log\det(Q_k^{-1/2}(x_k)).
\end{aligned} \tag{22}
$$

We now derive the explicit forms for C and a in (11) for the Gauss-Newton subproblem (10). Recall that C and a are given by

$$
\begin{array}{rcl}
C & = & [V\partial_x c + (c^{\mathrm{T}} \otimes I_N)\partial_x V]^{\mathrm{T}}[V\partial_x c + (c^{\mathrm{T}} \otimes I_N)\partial_x V], \\
a & = & c^{\mathrm{T}} V^{\mathrm{T}} V \partial_x c + c^{\mathrm{T}} V^{\mathrm{T}} (c^{\mathrm{T}} \otimes I_N)\partial_x V.
\end{array}
$$

where

$$
(c^{\mathrm{T}} \otimes I_N)\partial_x V \quad = \quad \sum_{i=1}^{M+nN} c_i \partial_x V_{i\cdot}.
$$

Define $\tilde{w}(x)$ and $\tilde{v}(x)$ in (3) by

$$
\tilde{w}_k(x) \quad = \quad x_k - g_k(x_{k-1}) \tag{23}
$$

$$
\tilde{v}_k(x) \quad = \quad z_k - h_k(x_k) \tag{24}
$$

In the expressions below, we will use notation $\tilde{w}_{k,i}$ to mean the ith component of $\tilde{w}_k$.

The matrix $(c^{\mathrm{T}} \otimes I_N)\partial_x V$ has the following block structure:

$$
[(c^{\mathrm{T}} \otimes I_N)\partial_x V] = \begin{bmatrix}
\tilde{Q}_1 & 0 & 0 & 0 & 0 & \cdots & 0 \\
\tilde{R}_1 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & \tilde{Q}_2 & 0 & 0 & 0 & \cdots & 0 \\
0 & \tilde{R}_2 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & \tilde{Q}_3 & 0 & 0 & \cdots & 0 \\
0 & 0 & \tilde{R}_3 & \ddots & 0 & \cdots & 0 \\
0 & 0 & 0 & \ddots & \ddots & \cdots & 0 \\
0 & 0 & 0 & \ddots & \ddots & Q_{N-1} & 0 \\
0 & 0 & 0 & \ddots & \ddots & R_{N-1} & 0 \\
0 & 0 & 0 & \cdots & \cdots & 0 & \tilde{Q}_N \\
0 & 0 & 0 & \cdots & \cdots & 0 & \tilde{R}_N
\end{bmatrix} \tag{25}
$$

where

$$
\tilde{Q}_i \quad = \quad \sum_{j=1}^{n} \tilde{w}_{i,j} \partial_{x_i} [Q_i^{-1/2}]_{j\cdot} \tag{26}
$$

$$
\tilde{R}_i \quad = \quad \sum_{j=1}^{m(i)} \tilde{v}_{i,j} \partial_{x_i} [R_i^{-1/2}]_{j\cdot}\ . \tag{27}
$$

Then we can write down the gradient a:

$$
a \quad = \quad \begin{bmatrix} a_1^{\mathrm{T}} & a_2^{\mathrm{T}} & \cdots & a_N^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \ ,
$$

where

$$
\begin{array}{rcl}
a_j & = & -\tilde{v}_j^{\mathrm{T}} R_j^{-1}(x_j)\partial_{x(j)} h_j(x_j) + \tilde{w}_j^{\mathrm{T}} Q_j^{-1}(x_{j-1}) + \tilde{w}_{j+1}^{\mathrm{T}} Q_{j+1}^{-1}(x_j)\partial_{x(j)} g_{j+1}(x_j) \\
 & & +\tilde{w}_j^{\mathrm{T}} Q_j^{-T/2} \tilde{Q}_j + \tilde{v}_j^{\mathrm{T}} R_j^{-T/2} \tilde{R}_j\ .
\end{array} \tag{28}
$$

We can compute C in a similar manner. First we write down the explicit form of $[V\partial_x c + (c^{\mathrm{T}} \otimes I_N)\partial_x V]$:

$$
\begin{bmatrix}
Q_1^{-1/2} + \tilde{Q}_1 & 0 & 0 & \cdots & 0 \\
-R_1^{-1/2}\nabla h_1 + \tilde{R}_1 & 0 & 0 & \cdots & 0 \\
-Q_2^{-1/2}\nabla g_2 & Q_2^{-1/2} + \tilde{Q}_2 & 0 & \cdots & 0 \\
0 & -R_2^{-1/2}\nabla h_2 + \tilde{R}_2 & \ddots & \cdots & 0 \\
0 & -Q_3^{-1/2}\nabla g_3 & \ddots & \cdots & 0 \\
0 & 0 & \ddots & Q_{N-1}^{-1/2} + \tilde{Q}_{N-1} & 0 \\
0 & 0 & 0 & -R_{N-1}^{-1/2}\nabla h_{N-1} + \tilde{R}_{N-1} & 0 \\
0 & 0 & 0 & -Q_{N-1}^{-1/2}\nabla g_N & Q_N^{-1/2} + \tilde{Q}_N \\
0 & 0 & 0 & 0 & -R_N^{-1/2}\nabla h_N + \tilde{R}_N
\end{bmatrix}. \tag{29}
$$

Multiplying the transpose of this matrix by itself, obtain the final form of C:

$$
\begin{array}{rcl}
C & = & \begin{bmatrix} C_1 & A_2^{\mathrm{T}} & 0 & \\ A_2 & C_2 & A_3^{\mathrm{T}} & 0 \\ 0 & \ddots & \ddots & \ddots \\ & 0 & A_N & C_N \end{bmatrix} \\
C_k & = & [Q_k^{-1/2} + \tilde{Q}_k]^{\mathrm{T}}[Q_k^{-1/2} + \tilde{Q}_k] + \nabla g_{k+1}^{\mathrm{T}} Q_{k+1}^{-1} \nabla g_{k+1} \\
 & + & [-R_k^{-1/2}\nabla h_k + \tilde{R}_k]^{\mathrm{T}}[-R_k^{-1/2}\nabla h_k + \tilde{R}_k] \\
A_k & = & -(Q_k^{-1/2} + \tilde{Q}_k)^{\mathrm{T}} Q_k^{-1/2} \nabla g_k \ .
\end{array} \tag{30}
$$

Remark 5.1 *The matrix (30) is block tridiagonal, and so it can be inverted with effort $O(n^3N)$ using the algorithm in [8].*

Recall the direction finding equation (19) in the Extended GN algorithm:

$$
\nabla F(s, \lambda, d) \begin{bmatrix} \Delta s \\ \Delta \lambda \\ \Delta d \end{bmatrix} = -F(s, \lambda, d).
$$

The solution to this system is given by (19):

$$
\begin{array}{rcl}
\Delta d & = & \left(C + \partial_x \mathrm{vec}\{V_{ii}\}^{\mathrm{T}} D(s)^{-1} D(\lambda) \partial_x \mathrm{vec}\{V_{ii}\}\right)^{-1} \gamma \\
\Delta \lambda & = & D(s)^{-1} \left[\mathbf{1} - D(\lambda)(\mathrm{vec}\{V_{ii}\} + \partial_x \mathrm{vec}\{V_{ii}\}(d + \Delta d))\right] \\
\Delta s & = & -s + \mathrm{vec}\{V_{ii}\} + \partial_x \mathrm{vec}\{V_{ii}\}(d + \Delta d),
\end{array}
$$

where

$$
\begin{array}{ll}
[\partial_x \mathrm{vec}\{V_{ii}\}^{\mathrm{T}} & D(s)^{-1} D(\lambda) \partial_x \mathrm{vec}\{V_{ii}\}]_k = \\
 & \partial_{x(k)} \mathrm{diag}\{Q_k^{-1/2}\}^{\mathrm{T}} D(s_{Q_k})^{-1} D(\lambda_{Q_k}) \partial_{x(k)} \mathrm{diag}\{Q_k^{-1/2}\} \\
 & + \partial_{x(k)} \mathrm{diag}\{R_k^{-1/2}\}^{\mathrm{T}} D(s_{R_k})^{-1} D(\lambda_{R_k}) \partial_{x(k)} \mathrm{diag}\{R_k^{-1/2}\} \ .
\end{array}
$$